UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/610,961	07/01/2003	Anand A. Kekre	VRT0063US	4162

60429 7590 06/23/2008
CAMPBELL STEPHENSON LLP
11401 CENTURY OAKS TERRACE
BLDG. H, SUITE 250
AUSTIN, TX 78758

EXAMINER	
DWIVEDI, MAHESH H	
ART UNIT	**PAPER NUMBER**
2168	
MAIL DATE	**DELIVERY MODE**
06/23/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/610,961	KEKRE ET AL.	
	Examiner	Art Unit	
	MAHESH H. DWIVEDI	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _15 May 2008_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1,4-13,15,18-26,30,31 and 33_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1,4-13,15,18-26,30,31 and 33_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _01 July 2003_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 9/5/2006 has been entered.

Remarks

2. Receipt of applicant's amendment filed on 01/30/2008 is acknowledged. The amendment includes the amending of claims 1 and 15, the addition of claim 33, and the cancellation of the claims 2-3, 14, 16-17, and 27-29, and 32.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1, 9, 15, 23, and 33 are rejected under 35 U.S.C. 102(e) as being anticipated by **Eshel et al.** (U.S. PGPUB 2003/0158862).

5. Regarding claim 1, **Eshel** teaches a method comprising:

A) <u>writing data to a first and second data volume</u> (Paragraphs 127, and 129-131);

B) <u>wherein the first data volume is a first primary volume</u> (Paragraph 130);

C) <u>the second data volume is a second primary volume</u> (Paragraph 130);

D) <u>the first primary volume and the second primary volumes are coupled to a host node</u> (Paragraphs 129-130);

E) <u>the host node processes requests received form at least one client computer to perform transactions on the first primary volume or the second primary volume</u> (Paragraphs 129-130);

F) refreshing the second data volume to the data contents of the first data volume that existed at time T (Paragraphs 127, 130);

G) wherein refreshing the second data volume comprises overwriting data of the second data volume with data of the first data volume that existed at time T (Paragraphs 127, 130);

H) modifying data of the first data volume while the second data volume is being refreshed to the data contents of the first data volume that existed at time T (Paragraph 127); and

I) modifying data of the first data volume after the second data volume has been refreshed (Paragraph 130).

The examiner notes that **Eshel** teaches **"<u>writing data to a first and second data volume</u>"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), "Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106" (Paragraph 129), and "Maintenance of the standby file system is facilitated in the exemplary embodiments by maintaining snapshot tags that uniquely identify both the different snapshots that recorded the state of each of the file systems at different times and that identify the set of changes that are generated between two snapshots. The snapshot tags are used to coordinate proper data synchronization between the mirror file system and the active file system when switching the mirror file system from a read only file system to the active read/write file system by ensuring that the latest snapshot is applied after a failure disables the original file system. Once the initial mirror file system becomes the active file system that is used by client processors (i.e., the "new original" file system), snapshots are captured of the new original file system and snapshot tags are used to restore the previous original file system, which is now the mirror, to maintain the original file system as the new standby, or mirror, file system"

(Paragraph 131). The examiner further notes that **Eshel** teaches **"wherein the first data volume is a first primary volume"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that **Eshel** teaches **"the second data volume is a second primary volume"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that **Eshel** teaches **"the first primary volume and the second primary volumes are coupled to a host node"** as "A block diagram of an overall system architecture for a primary and standby file system 1500 according to an exemplary embodiment of the present invention is illustrated in FIG. 15A. This exemplary system architecture has a primary file system, denoted as file system A 1502, a standby file system, denoted as file system B 1504 and a network 106 to provide communications between these file systems. Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106. File system A 1502 in this example has two snapshot datasets, a first snapshot dataset 1506 and a second snapshot dataset 1508. These two snapshot datasets captured the state of the file system A 1502 at different times. File system A 1502 operates by communicating snapshot datasets, such as first snapshot dataset 1506 and second snapshot 1508, to file system B 1504. File system B 1504, in turn, stores copies of the snapshot datasets that are received from file system A 1502. File system B 1504 stores a first snapshot dataset copy 1510 and a second snapshot dataset copy 1512 to support standby data storage operations. These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically

bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system. The original file system snapshot and the set of changes that are generated by these file systems contain tags to ensure completeness in the mirror file system by identifying the order of creation or the order in which these set of changes where applied. In this description, the term "restore" indicates a file system has been brought to the state of another file system by processing a dataset that represents an entire snapshot from that other file system. The term "apply" indicates that a file system has been updated to a more recent state of another file system by processing a set of changes that was generated between two snapshots on the other file system" (Paragraphs 129-130). The examiner further notes that **Eshel** teaches **"<u>the host node processes requests received form at least one client computer to perform transactions on the first primary volume or the second primary volume</u>"** as "A block diagram of an overall system architecture for a primary and standby file system 1500 according to an exemplary embodiment of the present invention is illustrated in FIG. 15A. This exemplary system architecture has a primary file system, denoted as file system A 1502, a standby file system, denoted as file system B 1504 and a network 106 to provide communications between these file systems. Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106. File system A 1502 in this example has two snapshot datasets, a first snapshot dataset 1506 and a second snapshot dataset 1508. These two snapshot datasets captured the state of the file system A 1502 at different times. File system A 1502 operates by communicating snapshot datasets, such as first snapshot dataset 1506 and second snapshot 1508, to file system B 1504. File system B 1504, in turn, stores copies of the snapshot datasets that are received from file system A 1502. File system B 1504 stores a first snapshot

dataset copy 1510 and a second snapshot dataset copy 1512 to support standby data storage operations. These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system. The original file system snapshot and the set of changes that are generated by these file systems contain tags to ensure completeness in the mirror file system by identifying the order of creation or the order in which these set of changes where applied. In this description, the term "restore" indicates a file system has been brought to the state of another file system by processing a dataset that represents an entire snapshot from that other file system. The term "apply" indicates that a file system has been updated to a more recent state of another file system by processing a set of changes that was generated between two snapshots on the other file system" (Paragraphs 129-130). The examiner further notes that **Eshel** teaches **"wherein the first data volume is unrelated to the second data volume in that the second data volume is not a point-in-time copy or a modified point-in-time copy of the first data volume"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that a "tape" is an external storage device that is

initially unrelated to the backup data it stores. Moreover, the examiner further wishes to state that the second file system of **Eshel** is also initially unrelated to the first file system. The examiner further notes that **Eshel** teaches **"refreshing the second data volume to the data contents of the first data volume that existed at time T"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that **Eshel** teaches **"wherein refreshing the second data volume comprises overwriting data of the second data volume with data of the first data volume that existed at time T"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that **Eshel** teaches **"modifying data of the first data volume while the second data volume is being refreshed to the data contents of the first data volume that existed at time T"** as Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127). The examiner further notes that **Eshel** teaches **"modifying data of the first data volume after the second data volume has been refreshed"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and

determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system. The original file system snapshot and the set of changes that are generated by these file systems contain tags to ensure completeness in the mirror file system by identifying the order of creation or the order in which these set of changes where applied. In this description, the term "restore" indicates a file system has been brought to the state of another file system by processing a dataset that represents an entire snapshot from that other file system. The term "apply" indicates that a file system has been updated to a more recent state of another file system by processing a set of changes that was generated between two snapshots on the other file system" (Paragraph 130).

Regarding claim 9, **Eshel** further teaches a method comprising:
A) wherein the second data volume is a real or virtual PIT copy of another data volume when the second data volume is refreshed to the data contents of the first data volume (Paragraph 127).

The examiner further notes that **Eshel** teaches **"wherein the second data volume is a real or virtual PIT copy of another data volume when the second data volume is refreshed to the data contents of the first data volume"** as Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127).

Regarding claim 15, **Eshel** teaches a computer readable medium comprising:
A) <u>writing data to a first and second data volume</u> (Paragraphs 127, and 129-131);
B) <u>wherein the first data volume is a first primary volume</u> (Paragraph 130);
C) <u>the second data volume is a second primary volume</u> (Paragraph 130);

D) the first primary volume and the second primary volumes are coupled to a host node (Paragraphs 129-130);

E) the host node processes requests received form at least one client computer to perform transactions on the first primary volume or the second primary volume (Paragraphs 129-130);

F) refreshing a second data volume to the data contents of the first data volume that existed at time T (Paragraphs 127, 130);

G) wherein refreshing the second data volume comprises overwriting data of the second data volume with data of the first data volume that existed at time T (Paragraphs 127, 130);

H) wherein the first data volume is unrelated to the second data volume prior to refreshing the second data volume to the data contents of the fists data volume (Paragraphs 127, 130);

I) modifying data of the first data volume while the second data volume is being refreshed to the data contents of the first data volume that existed at time T (Paragraph 127); and

J) modifying data of the first data volume after the second data volume has been refreshed (Paragraph 130).

The examiner notes that **Eshel** teaches **"writing data to a first and second data volume"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), "Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106" (Paragraph 129), and "Maintenance of the standby file system is facilitated in the exemplary embodiments by maintaining snapshot tags that uniquely identify both the different snapshots that recorded the state of each of the file systems at different times and that identify the set of changes that are generated between two snapshots. The snapshot tags are used to coordinate proper data synchronization between the mirror file system and the active file system when switching the mirror file system from a read only file system to the active read/write file system by ensuring that the latest snapshot

is applied after a failure disables the original file system. Once the initial mirror file system becomes the active file system that is used by client processors (i.e., the "new original" file system), snapshots are captured of the new original file system and snapshot tags are used to restore the previous original file system, which is now the mirror, to maintain the original file system as the new standby, or mirror, file system" (Paragraph 131). The examiner further notes that **Eshel** teaches **"wherein the first data volume is a first primary volume"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that **Eshel** teaches **"the second data volume is a second primary volume"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that **Eshel** teaches **"the first primary volume and the second primary volumes are coupled to a host node"** as "A block diagram of an overall system architecture for a primary and standby file system 1500 according to an exemplary embodiment of the present invention is illustrated in FIG. 15A. This exemplary system architecture has a primary file system, denoted as file system A 1502, a standby file system, denoted as file system B 1504 and a network 106 to provide communications between these file systems. Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106. File system A 1502 in this example has two snapshot datasets, a first snapshot dataset 1506 and a second snapshot dataset 1508. These two snapshot datasets captured the state of the file system A 1502 at different times. File system A 1502 operates by communicating snapshot datasets, such as first snapshot dataset 1506 and second snapshot 1508, to file system B 1504. File system B 1504, in turn, stores copies of the snapshot datasets that are received from file system A 1502. File system B 1504 stores a first snapshot dataset copy 1510 and a

second snapshot dataset copy 1512 to support standby data storage operations. These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system. The original file system snapshot and the set of changes that are generated by these file systems contain tags to ensure completeness in the mirror file system by identifying the order of creation or the order in which these set of changes where applied. In this description, the term "restore" indicates a file system has been brought to the state of another file system by processing a dataset that represents an entire snapshot from that other file system. The term "apply" indicates that a file system has been updated to a more recent state of another file system by processing a set of changes that was generated between two snapshots on the other file system" (Paragraphs 129-130). The examiner further notes that **Eshel** teaches **"<u>the host node processes requests received form at least one client computer to perform transactions on the first primary volume or the second primary volume</u>"** as "A block diagram of an overall system architecture for a primary and standby file system 1500 according to an exemplary embodiment of the present invention is illustrated in FIG. 15A. This exemplary system architecture has a primary file system, denoted as file system A 1502, a standby file system, denoted as file system B 1504 and a network 106 to provide communications between these file systems. Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106. File system A 1502 in this example has two snapshot datasets, a first snapshot dataset 1506 and a

second snapshot dataset 1508. These two snapshot datasets captured the state of the file system A 1502 at different times. File system A 1502 operates by communicating snapshot datasets, such as first snapshot dataset 1506 and second snapshot 1508, to file system B 1504. File system B 1504, in turn, stores copies of the snapshot datasets that are received from file system A 1502. File system B 1504 stores a first snapshot dataset copy 1510 and a second snapshot dataset copy 1512 to support standby data storage operations. These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system. The original file system snapshot and the set of changes that are generated by these file systems contain tags to ensure completeness in the mirror file system by identifying the order of creation or the order in which these set of changes where applied. In this description, the term "restore" indicates a file system has been brought to the state of another file system by processing a dataset that represents an entire snapshot from that other file system. The term "apply" indicates that a file system has been updated to a more recent state of another file system by processing a set of changes that was generated between two snapshots on the other file system" (Paragraphs 129-130). The examiner notes that **Eshel** teaches **"refreshing a second data volume to the data contents of the first data volume that existed at time T"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system

by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that **Eshel** teaches **"wherein refreshing the second data volume comprises overwriting data of the second data volume with data of the first data volume that existed at time T"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that **Eshel** teaches **"wherein the first data volume is unrelated to the second data volume prior to refreshing the second data volume to the data contents of the fists data volume"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that a "tape" is an external storage device that is initially unrelated to the backup data it stores. Moreover, the examiner further wishes to state that the second file system of **Eshel** is also initially unrelated to the first file system. The examiner further notes that **Eshel** teaches **"modifying data of the first data volume while the second data volume is being refreshed to the data contents of the first data volume that existed at time T"** as Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127). The examiner further notes that **Eshel** teaches **"modifying data of the first data volume after the second data volume has been refreshed"** as "These embodiments of the present invention create a hot standby

file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system. The original file system snapshot and the set of changes that are generated by these file systems contain tags to ensure completeness in the mirror file system by identifying the order of creation or the order in which these set of changes where applied. In this description, the term "restore" indicates a file system has been brought to the state of another file system by processing a dataset that represents an entire snapshot from that other file system. The term "apply" indicates that a file system has been updated to a more recent state of another file system by processing a set of changes that was generated between two snapshots on the other file system" (Paragraph 130).

Regarding claim 23, **Eshel** further teaches a computer readable medium comprising:

A) wherein the second data volume is a real or virtual PIT copy of another data volume when the second data volume is refreshed to the data contents of the first data volume (Paragraph 127).

The examiner further notes that **Eshel** teaches **"wherein the second data volume is a real or virtual PIT copy of another data volume when the second data volume is refreshed to the data contents of the first data volume"** as Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127).

Regarding claim 33, **Eshel** teaches a method comprising:

A) writing data to first and second data volumes (Paragraphs 127, and 129-131);

B) wherein the first data volume is unrelated to the second data volume in that the second data volume is not a point-in-time copy or a modified point-in-time copy of the first data volume (Paragraphs 127, 130); and

C) the first data volume is unrelated to the second data volume after the writing (Paragraphs 127, 130);

D) refreshing the second data volume to the data contents of the first data volume that existed at time T (Paragraphs 127, 130);

E) wherein refreshing the second data volume comprises overwriting all data of the second data volume with data of the first data volume that existed at time T (Paragraphs 127, 130);

F) modifying data of the first data volume while the second data volume is being refreshed to the data contents of the first data volume that existed at time T (Paragraph 127); and

G) modifying data of the first data volume after the second data volume has been refreshed (Paragraph 130).

The examiner notes that **Eshel** teaches **"writing data to first and second data volumes"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), "Alternative embodiments maintain the primary and backup file systems within a single processor, thereby obviating the requirement for a network 106" (Paragraph 129), and "Maintenance of the standby file system is facilitated in the exemplary embodiments by maintaining snapshot tags that uniquely identify both the different snapshots that recorded the state of each of the file systems at different times and that identify the set of changes that are generated between two snapshots. The snapshot tags are used to coordinate proper data synchronization between the mirror file system and the active file system when switching the mirror file system from a read only file system to the active read/write file system by ensuring that the latest snapshot

is applied after a failure disables the original file system. Once the initial mirror file system becomes the active file system that is used by client processors (i.e., the "new original" file system), snapshots are captured of the new original file system and snapshot tags are used to restore the previous original file system, which is now the mirror, to maintain the original file system as the new standby, or mirror, file system" (Paragraph 131). The examiner further notes that **Eshel** teaches **"wherein the first data volume is unrelated to the second data volume in that the second data volume is not a point-in-time copy or a modified point-in-time copy of the first data volume"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that a "tape" is an external storage device that is initially unrelated to the backup data it stores. Moreover, the examiner further wishes to state that the second file system of **Eshel** is also initially unrelated to the first file system. The examiner further notes that **Eshel** teaches **"the first data volume is unrelated to the second data volume after the writing"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that a "tape" is an external storage device that is initially unrelated to the backup data it stores. Moreover, the examiner further wishes to state that the second file system of **Eshel** is also initially unrelated to the first file system. Furthermore, the examiner wishes to state that the second file system is clearly operable to receive commands before being mirrored to the contents of the first file system. The examiner

further notes that **Eshel** teaches **"refreshing the second data volume to the data contents of the first data volume that existed at time T"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that **Eshel** teaches **"wherein refreshing the second data volume comprises overwriting all data of the second data volume with data of the first data volume that existed at time T"** as "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further notes that **Eshel** teaches **"modifying data of the first data volume while the second data volume is being refreshed to the data contents of the first data volume that existed at time T"** as Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127). The examiner further notes that **Eshel** teaches **"modifying data of the first data volume after the second data volume has been refreshed"** as "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system). These embodiments then periodically bring the standby or mirror file system up-to-date by generating new snapshots of the original file system and determining the changes between these new, more recently captured or generated snapshots and the state that was captured by a previous snapshot of the original file

system that had been transferred to the mirror file system. The original file system generates a set of changes that are then communicated and applied to the standby file system in order to bring the standby file system up to the state of the new snapshots captured on the original file system. The original file system snapshot and the set of changes that are generated by these file systems contain tags to ensure completeness in the mirror file system by identifying the order of creation or the order in which these set of changes where applied. In this description, the term "restore" indicates a file system has been brought to the state of another file system by processing a dataset that represents an entire snapshot from that other file system. The term "apply" indicates that a file system has been updated to a more recent state of another file system by processing a set of changes that was generated between two snapshots on the other file system" (Paragraph 130).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

8. Claims 4-5, 8, 10-12, 18-19, 22, and 24-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Eshel et al.** (U.S. PGPUB 2003/0158862) as applied to

claims 1, 9, 15, 23, and 33 above, and in view of **Veritas** (Article entitled "Veritas Flashsnap Point-in-Time Copy Solutions", dated 06/24/2002).

9. Regarding claim 4, **Eshel** does not explicitly teach a method comprising:

A) creating one or more PIT copies of the first data volume prior to refreshing the second data volume to the data contents of the first data volume.

Veritas, however, teaches **"creating one or more PIT copies of the first data volume prior to refreshing the second data volume to the data contents of the first data volume"** as "1. Create snapshot mirrors: Use vxassist snapstart to create snapshot mirrors of one or more volumes" (Page 10, Section: Implementing Point-in Time Copy Solutions on a Primary Host).

The examiner notes that it is clear that **Veritas** creates multiple mirrors of primary volumes before refreshing the primary volume onto a secondary volume.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 5, **Eshel** does not explicitly teach a method comprising:

A) wherein one of the PIT copies of the first data volume is in a virtual state when the second data volume is refreshed to the data contents of the first data volume.

Veritas, however, teaches **"wherein one of the PIT copies of the first data volume is in a virtual state when the second data volume is refreshed to the data contents of the first data volume"** as "The presence of the FastResync map means that only those updates that the mirror has missed need to be reapplied to resynchronize it with the volume. A full, and thereby much slower, resynchronization of the mirror form the volume is unnecessary" (Page 7, Section: FastResync of Volume Snapshots).

The examiner notes that it is clear that **Veritas's** snapshot mirrors are virtual in that they contain data stored in the primary volume (see only updated data is migrated

to the mirror for resynchronization). The examiner further notes that it is common knowledge that Flashsnap creates virtual point-in-time copies of volumes.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 8, **Eshel** does not explicitly teach a method comprising:
A) wherein the first data volume is a real or virtual PIT copy of another data volume when the second data volume is refreshed to the data contents of the first data volume.

Veritas, however, teaches **"wherein the first data volume is a real or virtual PIT copy of another data volume when the second data volume is refreshed to the data contents of the first data volume"** as "1. Create snapshot mirrors: Use vxassist snapstart to create snapshot mirrors of one or more volumes...Use vxassist snapshot to create snapshot volumes from the snapshot mirrors" (Page 10, Section: Implementing Point-in Time Copy Solutions on a Primary Host).

The examiner notes that it is clear that **Veritas** has the snapshot volume refreshed to the state of the snapshot mirror, wherein the snapshot mirror is a point-in-time copy of the volume.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 10, **Eshel** does not explicitly teach a method comprising:
A) generating first and second maps in memory;
B) wherein each of the first and second maps comprises a plurality of entries;

C) wherein each entry of the first map corresponds to a respective memory block that stores data of the first data volume; and

D) wherein each entry of the second map corresponds to a respective memory block that stores data of the second data volume.

Veritas, however, teaches **"generating first and second maps in memory"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots), **"wherein each of the first and second maps comprises a plurality of entries"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots), **"wherein each entry of the first map corresponds to a respective memory block that stores data of the first data volume"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots), and **"wherein each entry of the second map corresponds to a respective memory block that stores data of the second data volume"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 11, **Eshel** does not explicitly teach a method comprising:

A) setting a first bit in each entry of the first map, wherein each first bit of the first map is set to indicate its respective memory block stores valid data;
B) clearing a first bit in each entry of the second map, wherein each first bit of the second map is set to indicate its respective memory block stores invalid data.

Veritas, however, teaches **"setting a first bit in each entry of the first map, wherein each first bit of the first map is set to indicate its respective memory block stores valid data"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots), and **"clearing a first bit in each entry of the second map, wherein each first bit of the second map is set to indicate its respective memory block stores invalid data"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots).

The examiner notes that it is clear that **Veritas's** maps have a plurality of entries and track changes to both the primary volume and the snapshot volume (see "keep track of which blocks are updated in the volume and in the snapshot").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 12, **Eshel** does not explicitly teach a method comprising:
A) setting or clearing a second bit in each entry of the second map to indicate that its respective memory block stores data needed for a PIT copy of the second data volume.

Veritas, however, teaches **"setting or clearing a second bit in each entry of the second map to indicate that its respective memory block stores data needed**

for a PIT copy of the second data volume" as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots).

The examiner notes that it is clear that **Veritas's** maps have a plurality of entries and track changes to both the primary volume and the snapshot volume (see "keep track of which blocks are updated in the volume and in the snapshot").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 18, **Eshel** does not explicitly teach a computer readable medium comprising:

A) wherein the method further comprises creating one or more PIT copies of the first data volume prior to refreshing the second data volume to the data contents of the first data volume.

Veritas, however, teaches **"wherein the method further comprises creating one or more PIT copies of the first data volume prior to refreshing the second data volume to the data contents of the first data volume"** as "1. Create snapshot mirrors: Use vxassist snapstart to create snapshot mirrors of one or more volumes" (Page 10, Section: Implementing Point-in Time Copy Solutions on a Primary Host).

The examiner notes that it is clear that **Veritas** creates multiple mirrors of primary volumes before refreshing the primary volume onto a secondary volume.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 19, **Eshel** does not explicitly teach a computer readable medium comprising:

A) wherein one of the PIT copies of the first data volume is in a virtual state when the second data volume is refreshed to the data contents of the first data volume.

Veritas, however, teaches **"wherein one of the PIT copies of the first data volume is in a virtual state when the second data volume is refreshed to the data contents of the first data volume"** as "The presence of the FastResync map means that only those updates that the mirror has missed need to be reapplied to resynchronize it with the volume. A full, and thereby much slower, resynchronization of the mirror form the volume is unnecessary" (Page 7, Section: FastResync of Volume Snapshots).

The examiner notes that it is clear that **Veritas's** snapshot mirrors are virtual in that they contain data stored in the primary volume (see only updated data is migrated to the mirror for resynchronization). The examiner further notes that it is common knowledge that Flashsnap creates virtual point-in-time copies of volumes.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 22, **Eshel** does not explicitly teach a computer readable medium comprising:

A) wherein the first data volume is a real or virtual PIT copy of another data volume when the second data volume is refreshed to the data contents of the first data volume.

Veritas, however, teaches **"wherein the first data volume is a real or virtual PIT copy of another data volume when the second data volume is refreshed to the data contents of the first data volume"** as "1. Create snapshot mirrors: Use vxassist snapstart to create snapshot mirrors of one or more volumes…Use vxassist snapshot to

create snapshot volumes from the snapshot mirrors" (Page 10, Section: Implementing Point-in Time Copy Solutions on a Primary Host).

The examiner notes that it is clear that **Veritas** has the snapshot volume refreshed to the state of the snapshot mirror, wherein the snapshot mirror is a point-in-time copy of the volume.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 24, **Eshel** does not explicitly teach a computer readable medium comprising:

A) wherein refreshing the second data volume further comprises generating first and second maps in memory;

B) wherein each of the first and second maps comprises a plurality of entries;

C) wherein each entry of the first map corresponds to a respective memory block that stores data of the first data volume; and

D) wherein each entry of the second map corresponds to a respective memory block that stores data of the second data volume.

Veritas, however, teaches **"wherein refreshing the second data volume further comprises generating first and second maps in memory"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots), **"wherein each of the first and second maps comprises a plurality of entries"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots), **"wherein each entry**

of the first map corresponds to a respective memory block that stores data of the first data volume" as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots), and **"wherein each entry of the second map corresponds to a respective memory block that stores data of the second data volume"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots).

The examiner notes that it is clear that **Veritas's** maps have a plurality of entries and track changes to both the primary volume and the snapshot volume (see "keep track of which blocks are updated in the volume and in the snapshot").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 25, **Eshel** does not explicitly teach a computer readable medium comprising:

A) clearing a first bit in each entry of- the first map, wherein each first bit of the first map is set to indicate its respective memory block stores valid data;

B) setting a first bit in each entry of the second map, wherein each first bit of the second map is set to indicate its respective memory block stores invalid data.

Veritas, however, teaches **"clearing a first bit in each entry of- the first map, wherein each first bit of the first map is set to indicate its respective memory block stores valid data"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in

memory" (Page 7, Section: FastResync of Volume Snapshots), and **"setting a first bit in each entry of the second map, wherein each first bit of the second map is set to indicate its respective memory block stores invalid data"** as "VxVM uses a FastResync map to keep track of which blocks are updated in the volume and in the snapshot" (Page 7, Section: FastResync of Volume Snapshots) and "Non=-Persistent FastResync stores its change maps in memory" (Page 7, Section: FastResync of Volume Snapshots).

The examiner notes that it is clear that **Veritas's** maps have a plurality of entries and track changes to both the primary volume and the snapshot volume (see "keep track of which blocks are updated in the volume and in the snapshot").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

10. Claims 6-7, 13, 20-21, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Eshel et al.** (U.S. PGPUB 2003/0158862) as applied to claims 1, 9, 15, 23, and 33 above, and in view of **Veritas** (Article entitled "Veritas Flashsnap Point-in-Time Copy Solutions", dated 06/24/2002) as applied to claims 4-5, 8, 10-12, 18-19, 22, and 24-25, and further in view of **DeKoning** (U.S. Patent 6,691,245).

11. Regarding claim 6, **Eshel** and **Veritas** do not explicitly teach a method comprising:

A) further comprising an act of preserving the second data volume, wherein said preserving comprises creating one or more PIT copies of the second data volume prior to refreshing the second data volume to the data contents of the first data volume.

DeKoning, however, teaches **"further comprising an act of preserving the second data volume, wherein said preserving comprises creating one or more PIT copies of the second data volume prior to refreshing the second data volume to the data contents of the first data volume"** as "An incremental snapshot of the mirrored data is generated on the secondary storage device at the predetermined

checkpoint indicated by the checkpoint message…Thus, the incremental snapshot maintains the storage state of the secondary storage device at the predetermined checkpoint" (Column 2, lines 59-67-Column 3, lines 10).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **DeKoning's** would have allowed **Eshel's** and **Veritas's** to improve efficiency in dealing with synchronization through volume preservation, as noted by **DeKoning** (Column 2, lines 1-5).

Regarding claim 7, **Eshel** does not explicitly teach a method comprising:
A) wherein one of the PIT copies of the second data volume is in the virtual state when the second data volume is refreshed to the data contents of the first data volume.

Veritas, however, teaches **"wherein one of the PIT copies of the second data volume is in the virtual state when the second data volume is refreshed to the data contents of the first data volume"** as "1. Create snapshot mirrors: Use vxassist snapstart to create snapshot mirrors of one ore more volumes…Use vxassist snapshot to create snapshot volumes from the snapshot mirrors" (Page 10, Section: Implementing Point-in Time Copy Solutions on a Primary Host).

The examiner notes that it is clear that **Veritas** has the snapshot volume refreshed to the state of the snapshot mirror, wherein the snapshot mirror is a point-in-time copy of the volume.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 13, **Eshel** and **Veritas** do not explicitly teach a method comprising:

A) further comprising an act of preserving the second data volume, wherein said preserving comprises creating a PIT copy of the second data volume before or while refreshing the second data volume to the data contents of the first data volume.

DeKoning, however, teaches **"further comprising an act of preserving the second data volume, wherein said preserving comprises creating a PIT copy of the second data volume before or while refreshing the second data volume to the data contents of the first data volume"** as "An incremental snapshot of the mirrored data is generated on the secondary storage device at the predetermined checkpoint indicated by the checkpoint message...Thus, the incremental snapshot maintains the storage state of the secondary storage device at the predetermined checkpoint" (Column 2, lines 59-67-Column 3, lines 10).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **DeKoning's** would have allowed **Eshel's** and **Veritas's** to improve efficiency in dealing with synchronization through volume preservation, as noted by **DeKoning** (Column 2, lines 1-5).

Regarding claim 20, **Eshel** and **Veritas** do not explicitly teach a computer readable medium comprising:

A) further comprising an act of preserving the second data volume, wherein said preserving further comprises creating one or more PIT copies of the second data volume prior to refreshing the second data volume to the data of the first data volume.

DeKoning, however, teaches **"wherein said preserving further comprises creating one or more PIT copies of the second data volume prior to refreshing the second data volume to the data of the first data volume"** as "An incremental snapshot of the mirrored data is generated on the secondary storage device at the predetermined checkpoint indicated by the checkpoint message...Thus, the incremental snapshot maintains the storage state of the secondary storage device at the predetermined checkpoint" (Column 2, lines 59-67-Column 3, lines 10).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **DeKoning's** would have allowed **Eshel's** and **Veritas's** to improve efficiency in dealing with synchronization through volume preservation, as noted by **DeKoning** (Column 2, lines 1-5).

Regarding claim 21, **Eshel** does not explicitly teach a computer readable medium comprising:

A) wherein one of the PIT copies of the second data volume is in the virtual state when the second data volume is refreshed to the data contents of the first data volume.

Veritas, however, teaches **"wherein one of the PIT copies of the second data volume is in the virtual state when the second data volume is refreshed to the data contents of the first data volume"** as "1. Create snapshot mirrors: Use vxassist snapstart to create snapshot mirrors of one ore more volumes…Use vxassist snapshot to create snapshot volumes from the snapshot mirrors" (Page 10, Section: Implementing Point-in Time Copy Solutions on a Primary Host).

The examiner notes that it is clear that **Veritas** has the snapshot volume refreshed to the state of the snapshot mirror, wherein the snapshot mirror is a point-in-time copy of the volume.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Veritas's** would have allowed **Eshel's** to provide a method to improve efficiency in resynchronization by applying changes to only the updates a mirror has missed, as noted by **Veritas** (Page 7, Section: FastResync of Volume Snapshots).

Regarding claim 26, **Eshel** and **Veritas** do not explicitly teach a computer readable medium comprising:

A) further comprising an act of preserving the second data volume, wherein said preserving further comprises creating a PIT copy of the second data volume before or while refreshing the second data volume to the data of the first data volume.

DeKoning, however, teaches **"further comprising an act of preserving the second data volume, wherein said preserving further comprises creating a PIT copy of the second data volume before or while refreshing the second data volume to the data of the first data volume"** as "An incremental snapshot of the mirrored data is generated on the secondary storage device at the predetermined checkpoint indicated by the checkpoint message…Thus, the incremental snapshot maintains the storage state of the secondary storage device at the predetermined checkpoint" (Column 2, lines 59-67-Column 3, lines 10).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **DeKoning's** would have allowed **Eshel's** and **Veritas's** to improve efficiency in dealing with synchronization through volume preservation, as noted by **DeKoning** (Column 2, lines 1-5).

12. Claims 30-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Eshel et al.** (U.S. PGPUB 2003/0158862) as applied to claims 1, 9, 15, 23, and 33 above, and in view of **Veritas** (Article entitled "Veritas Flashsnap Point-in-Time Copy Solutions", dated 06/24/2002) as applied to claims 4-5, 8, 10-2, 18-19, 22, and 24-25, and further in view of **Rand** (U.S. PGPUB 2005/0108302).

13. Regarding claim 30, **Eshel** and **Veritas** do not explicitly teach a method comprising:

A) further comprising an act of preserving the second data volume, wherein said preserving further comprises creating a PIT copy of the second data volume before or while refreshing the second data volume to the data of the first data volume; and

B) wherein, in response to the modifying the second data volume, the second data volume becomes a modified point-in-time copy of the first data volume that existed at time T.

Rand, however, teaches **"modifying data of the second data volume while the second data volume is being refreshed to the data contents of the first data volume that existed at time T"** as "FIG. 6 depicts in more detail an exemplary process 600 of satisfying read and write requests to primary data volume 112 while the primary

data volume 112 is being restored. In step 602, a determination is made as to whether the data storage drive having the primary data volume is active. If the data storage device is not active, then in step 604, the read/write requests to the primary data volume are satisfied using the generated image of the primary data volume. If the data storage device is active, then in step 606, a determination is made as to whether the request is a write request. If the request is a write request, then in step 608 the write request is satisfied by the primary data volume" (Paragraph 35) and **"wherein, in response to the modifying the second data volume, the second data volume becomes a modified point-in-time copy of the first data volume that existed at time T"** as "FIG. 6 depicts in more detail an exemplary process 600 of satisfying read and write requests to primary data volume 112 while the primary data volume 112 is being restored. In step 602, a determination is made as to whether the data storage drive having the primary data volume is active. If the data storage device is not active, then in step 604, the read/write requests to the primary data volume are satisfied using the generated image of the primary data volume. If the data storage device is active, then in step 606, a determination is made as to whether the request is a write request. If the request is a write request, then in step 608 the write request is satisfied by the primary data volume" (Paragraph 35).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Rand's** would have allowed **Eshel's** and **Veritas's** to allow for read/write requests during restoration of volumes, as noted by **Rand** (Paragraph 6).

Regarding claim 30, **Eshel** and **Veritas** do not explicitly teach a computer readable medium comprising:

A) further comprising an act of preserving the second data volume, wherein said preserving further comprises creating a PIT copy of the second data volume before or while refreshing the second data volume to the data of the first data volume; and

B) wherein, in response to the modifying the second data volume, the second data volume becomes a modified point-in-time copy of the first data volume that existed at time T.

Rand, however, teaches **"modifying data of the second data volume while the second data volume is being refreshed to the data contents of the first data volume that existed at time T"** as "FIG. 6 depicts in more detail an exemplary process 600 of satisfying read and write requests to primary data volume 112 while the primary data volume 112 is being restored. In step 602, a determination is made as to whether the data storage drive having the primary data volume is active. If the data storage device is not active, then in step 604, the read/write requests to the primary data volume are satisfied using the generated image of the primary data volume. If the data storage device is active, then in step 606, a determination is made as to whether the request is a write request. If the request is a write request, then in step 608 the write request is satisfied by the primary data volume" (Paragraph 35) and **"wherein, in response to the modifying the second data volume, the second data volume becomes a modified point-in-time copy of the first data volume that existed at time T"** as "FIG. 6 depicts in more detail an exemplary process 600 of satisfying read and write requests to primary data volume 112 while the primary data volume 112 is being restored. In step 602, a determination is made as to whether the data storage drive having the primary data volume is active. If the data storage device is not active, then in step 604, the read/write requests to the primary data volume are satisfied using the generated image of the primary data volume. If the data storage device is active, then in step 606, a determination is made as to whether the request is a write request. If the request is a write request, then in step 608 the write request is satisfied by the primary data volume" (Paragraph 35).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Rand's** would have allowed **Eshel's** and **Veritas's** to allow for read/write requests during restoration of volumes, as noted by **Rand** (Paragraph 6).

Response to Arguments

14. Applicant's arguments filed 05/15/2008 have been fully considered but they are not persuasive.

Applicant argues on page 11 that **"The system and method disclosed in Eshel frequently refer to "primary" and "backup" file systems. In fact paragraph [0127] of Eshel discloses backing up a file system to tape while allowing read/write access to the file system. Clearly, the tape used for a backup and not as a primary volume that processes transactions from a host node in response to the host node receiving requests from a client computer"**. However, the examiner wishes to refer to paragraphs 127, 129, and 130 of **Eshel** which state "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source) file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). Moreover, the limitation from the independent claims merely states that "<u>the host node processes requests received form at least one client computer to perform transactions on the first primary volume **or** the second primary volume</u>". Moreover, because the aforementioned limitation merely requires that <u>either</u> the first primary or second primary volume process transactions, then **Eshel's** method of processing the read/write requests to the first file system teaches the aforementioned limitation.

Applicant argues on page 11 that **"The fact that the tape may at one time be unrelated to a file system is irrelevant, since the tape disclosed in Eshel is clearly discloses as a backup volume rather than a primary volume, as recited in the independent claims"**. However, the examiner wishes to refer to paragraphs 127, 129, and 130 of **Eshel** which state "Another common use of snapshots is to back up a file system to tape while allowing continued read/write access to the file system during the backup process" (Paragraph 127), and "These embodiments of the present invention create a hot standby file system by first generating a snapshot of the original (source)

file system and transferring the entire data set for that snapshot to a second file system in order to create an identical copy of the original file system (i.e., a mirror file system)" (Paragraph 130). The examiner further wishes to state that the second file system of **Eshel** clearly teaches a primary volume since it eventually becomes a mirror of the first file system.

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent 6,665,815 issued to **Goldstein et al.** on 16 December 2003. The subject matter disclosed therein is pertinent to that of claims 1, 4-13, 15, 18-26, 30-31, and 33 (e.g., methods to spur synchronization via snapshots amongst varied data volumes).

U.S. Patent 6,611,901 issued to **Micka et al.** on 26 August 2003. The subject matter disclosed therein is pertinent to that of claims 1, 4-13,15, 18-26, 30-31, and 33 (e.g., methods to spur synchronization via snapshots amongst varied data volumes).

U.S. Patent 6,799,258 issued to **Linde et al.** on 28 September 2004. The subject matter disclosed therein is pertinent to that of claims 1, 4-13, 15, 18-26, 30-31, and 33 (e.g., methods to spur synchronization via snapshots amongst varied data volumes).

U.S. Patent 5,875,479 issued to **Blount et al.** on 23 February 1999. The subject matter disclosed therein is pertinent to that of claims 1, 4-13, 15, 18-26, 30-31, and 33 (e.g., methods to spur synchronization via snapshots amongst varied data volumes).

U.S. Patent 6,338,114 issued to **Paulsen et al.** on 08 January 2002. The subject matter disclosed therein is pertinent to that of claims 1, 4-13, 15, 18-26, 30-31, and 33 (e.g., methods to spur synchronization via snapshots amongst varied data volumes).

Article entitled "VERITAS FlashSnap: Using VERITAS FlashSnap to Protect Application Performance and Availability, by: **VERITAS**, dated 05/14/2002. The subject matter disclosed therein is pertinent to that of claims 1, 4-13, 15, 18-26, 30-31, and 33 (e.g., methods to spur synchronization via snapshots amongst varied data volumes).

Article entitled "VERITAS FlashSnap: Guidelines for Using VERITAS FlashSnap, by: **VERITAS**, dated 05/01/2002. The subject matter disclosed therein is pertinent to that of claims 1, 4-13, 15, 18-26, 30-31, and 33 (e.g., methods to spur synchronization via snapshots amongst varied data volumes).

U.S. Patent 7,085,901 issued to **Homma et al.** on 01 August 2006. The subject matter disclosed therein is pertinent to that of claims 1, 4-13, 15, 18-26, 30-31, and 33 (e.g., methods to spur synchronization via snapshots amongst varied data volumes).

U.S. Patent 6,643,671 issued to **Milillo et al.** on 04 November 2003. The subject matter disclosed therein is pertinent to that of claims 1, 4-13, 15, 18-26, 30-31, and 33 (e.g., methods to spur synchronization via snapshots amongst varied data volumes).

Article entitled "Shadow Copied of Shared Folders: Frequently Asked Questions", by: **Microsoft**, dated 03/03/2003. The subject matter disclosed therein is pertinent to that of claims 1, 4-13, 15, 18-26, 30-31, and 33 (e.g., methods to spur synchronization via snapshots amongst varied data volumes).

Contact Information

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi

Patent Examiner
Art Unit 2168

June 18, 2008
/Mahesh H Dwivedi/
Examiner, Art Unit 2168

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168